

## CLAIMS

What is claimed is:

1. In a computer system, a method for implementing an execution stack that  
5 stores frames for functions written in a plurality of programming languages, the method comprising:
- storing a first frame on the execution stack for a first function, the first function being written in a first programming language; and
- in response to the first function calling a second function written in a second  
10 programming language, storing a data block on the execution stack before a second frame for the second function, the data block including at least one pointer to a previous frame on the execution stack for a previous function written in the second programming language.
- 15 2. The method of claim 1, wherein the at least one pointer includes a previous stack pointer and frame pointer.
3. The method of claim 1, further comprising in response to the first function calling the second function, allocating resources for functions written in programming  
20 languages other than the second programming language that may be called by the second function.
4. The method of claim 3, further comprising upon exiting the second  
25 function, deallocating the resources for functions written in programming languages other than the second programming language.

5. The method of claim 1, further comprising catching an exception that was raised during execution of the second function that was not handled by an exception handler for the second function.

6. The method of claim 5, further comprising identifying an exception handler for the data block to handle the exception and jumping to the identified exception handler.

7. The method of claim 6, wherein the identified exception handler stores the exception in local storage.

8. The method of claim 7, wherein the local storage is storage associated with a current thread in which the first and second functions are executing.

9. The method of claim 7, further comprising upon returning to the first function, checking the local storage to determine if an exception is pending and throwing the stored exception if an exception is pending.

10. The method of claim 9, further comprising converting the stored exception to a format for the first programming language.

<sup>3</sup>  
~~11.~~ The method of claim 1, wherein the second programming language is the Java programming language.

Sub  
a1 → 12. A computer program product that implements an execution stack that stores frames for functions written in a plurality of programming languages, comprising:  
computer code that stores a first frame on the execution stack for a first function, the first function being written in a first programming language;  
computer code that, in response to the first function calling a second function written in a second programming language, stores a data block on the execution stack

the execution stack and storing a second frame on the execution stack for the second function.

8  
16 The method of claim 7, wherein the at least one pointer includes a  
5 previous stack pointer and frame pointer.

9  
17 The method of claim 7, wherein the local storage is storage associated  
with a current thread in which the first and second functions are executing.

10 10  
18 The method of claim 7, further comprising upon exiting the second  
function, clearing the at least one pointer stored in the local storage.

11  
19 The method of claim 7, further comprising catching an exception that was  
raised during execution of the second function that was not handled by an exception  
15 handler for the second function.

12  
20 The method of claim 11, further comprising determining if the exception is  
appropriate for the first programming language.

13  
21 The method of claim 11, further comprising storing the exception in the  
local storage.

14  
22 The method of claim 11, further comprising patching a return address on  
the execution stack with an address of an exception forwarder, the exception forwarder  
25 identifying an exception handler for the first function to handle the exception and jumping  
to the identified exception handler.

15  
23 The method of claim 14, wherein the exception forwarder converts the  
exception to a format for the first programming language.

0894435-10059

<sup>16</sup>  
~~24.~~ The method of claim <sup>7</sup>~~15~~, further comprising in response to the second function calling a third function written in the first programming language, storing a data block on the execution stack before a third frame for the third function, the data block including the at least one pointer to the first frame that is stored in the local storage.

<sup>17</sup>  
~~25.~~ The method of claim <sup>16</sup>~~24~~, further comprising in response to the second function calling the third function, allocating resources for functions written in programming languages other than the first programming language that may be called by the third function.

<sup>18</sup>  
~~26.~~ The method of claim <sup>17</sup>~~25~~, further comprising upon exiting the third function, deallocating the resources for functions written in programming languages other than the first programming language.

<sup>19</sup>  
~~27.~~ The method of claim <sup>18</sup>~~26~~, further comprising catching an exception that was raised during execution of the third function that was not handled by an exception handler for the third function.

<sup>20</sup>  
~~28.~~ The method of claim <sup>19</sup>~~27~~, further comprising identifying an exception handler for the data block to handle the exception and jumping to the identified exception handler.

<sup>21</sup>  
~~29.~~ The method of claim <sup>20</sup>~~28~~, wherein the identified exception handler stores the exception in the local storage.

<sup>22</sup>  
~~30.~~ The method of claim <sup>21</sup>~~29~~, further comprising upon returning to the second function, checking the local storage to determine if an exception is pending and throwing the stored exception if an exception is pending.

23  
31.

22

The method of claim 30, further comprising converting the stored exception to a format for the second programming language.

5

24  
32.

7

The method of claim 15, wherein the first programming language is the Java programming language.

Sub  
a2

33. A computer program product that implements an execution stack that stores frames for functions written in a plurality of programming languages, comprising:

10

computer code that stores a first frame on the execution stack for a first function, the first function being written in a first programming language;

computer code that, in response to the first function calling a second function written in a second programming language, stores in local storage at least one pointer to the first frame on the execution stack and stores a second frame on the execution stack for the second function; and

15

a computer readable medium that stores the computer codes.

34. The computer program product of claim 33, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

20

35. A computer system for implementing an execution stack that stores frames for functions written in a plurality of programming languages, comprising:

a processor;

25

a memory coupled to the processor that stores the execution stack; and

an computer program operating on the processor that stores a first frame on the execution stack for a first function, the first function being written in a first programming language; and, in response to the first function calling a second function written in a second programming language, stores in local storage at least one pointer to the first frame

08944335, 100697

on the execution stack and stores a second frame on the execution stack for the second function.

36. A data structure stored by a computer readable medium for implementing  
5 an execution stack, comprising:

a first frame stored by the computer readable medium on the execution stack, the first frame being for a first function written in a first programming language;

a second frame stored by the computer readable medium on the execution stack above the first frame, the second frame being for a second function written in a second  
10 programming language; and

a data block stored by the computer readable medium on the execution stack above the second frame, the data block including at least one pointer to the first frame on the execution stack.

15 <sup>29</sup>  
~~37.~~ The data structure of claim <sup>28</sup>~~36~~, wherein the at least one pointer includes a stack pointer and frame pointer to the first frame on the execution stack.

<sup>30</sup>  
~~38.~~ The data structure of claim <sup>28</sup>~~36~~, further comprising a return address stored by the computer readable medium on the execution stack between the first frame and the  
20 second frame, the return address being for the first function.

<sup>31</sup>  
~~39.~~ The data structure of claim <sup>28</sup>~~36~~, further comprising a return address stored by the computer readable medium on the execution stack between the second frame and the data block, the return address being for the second function.

25 <sup>32</sup>  
~~40.~~ The data structure of claim <sup>28</sup>~~36~~, wherein the first frame stores data selected from the group consisting of state variables, local variables and operand stack.

33

28

~~41.~~ The data structure of claim ~~36~~, wherein the first programming language is the Java programming language.

add  
a3

08944333 100597

30

before a second frame for the second function, the data block including at least one pointer to a previous frame on the execution stack for a previous function written in the second programming language; and

a computer readable medium that stores the computer codes.

5

13. The computer program product of claim 12, wherein the computer readable medium is selected from the group consisting of CD-ROM, floppy disk, tape, flash memory, system memory, hard drive, and data signal embodied in a carrier wave.

10

14. A computer system for implementing an execution stack that stores frames for functions written in a plurality of programming languages, comprising:

a processor;

a memory coupled to the processor that stores the execution stack; and

a computer program operating on the processor that stores a first frame on the

15

execution stack for a first function, the first function being written in a first programming language and, in response to the first function calling a second function written in a second programming language, stores a data block on the execution stack before a second frame for the second function, the data block including at least one pointer to a previous frame on the execution stack for a previous function written in the second programming language.

20

15. In a computer system, a method for implementing an execution stack that stores frames for functions written in a plurality of programming languages, the method comprising:

25

storing a first frame on the execution stack for a first function, the first function being written in a first programming language; and

in response to the first function calling a second function written in a second programming language, storing in local storage at least one pointer to the first frame on

08944335, 1006937, 669007, 5E4H680